



INSTITUTO
FEDERAL
Rio Grande
do Sul

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Upload de
Arquivos

Desenvolvimento de Aplicativo WEB I

Prof. Me. Cleber Schroeder Fonseca

Curso de Tecnologia em Análise e Desenvolvimento de Sistemas
IFRS Rio Grande

2024

① Upload de Arquivos

① Upload de Arquivos

Upload de arquivos é o processo de enviar dados de um dispositivo local para um servidor remoto. É uma funcionalidade comum em muitos sites e aplicações web, permitindo aos usuários compartilhar arquivos, como imagens, documentos, vídeos e muito mais.

Upload de Arquivos

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Upload de
Arquivos

O upload de arquivos é importante por várias razões:

- **Compartilhamento de Informações:** Permite aos usuários compartilhar informações de maneira eficiente e rápida. Por exemplo, um usuário pode fazer upload de um documento para compartilhá-lo com outros usuários em uma plataforma de colaboração.
- **Armazenamento e Backup:** Os usuários podem fazer upload de arquivos para armazená-los em um servidor remoto como uma forma de backup. Isso ajuda a proteger os dados contra perda em caso de falha do dispositivo local.
- **Funcionalidade do Site:** Muitos sites dependem do upload de arquivos para fornecer funcionalidades essenciais. Por exemplo, sites de redes sociais permitem que os usuários façam upload de fotos e vídeos.
- **Colaboração:** Em ambientes de trabalho, o upload de arquivos permite a colaboração em documentos e outros arquivos, facilitando o trabalho em equipe.

O que é o Upload de Arquivos?

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Upload de
Arquivos

- Upload de arquivos é o processo de transferência de um arquivo de um dispositivo local (como um computador ou celular) para um sistema remoto, como um servidor. Isso permite que uma cópia do arquivo seja armazenada em um local distante.
- Por exemplo, quando você envia um arquivo por e-mail, posta uma foto no Instagram, ou publica um vídeo no YouTube, você está realizando um upload. Essas ações dependem da troca de informações entre dois ou mais computadores.
- É importante notar que cada site ou rede social pode ter processos diferentes para receber arquivos. Mas, de maneira geral, o processo de upload é simples: você seleciona o arquivo desejado em seu dispositivo, aguarda o servidor receber os dados e confirma o upload.

PHP e Upload de Arquivos

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Upload de
Arquivos

- O PHP oferece suporte integral para o upload de arquivos, permitindo que você controle completamente quem pode fazer o upload e o que deve ser feito com o arquivo após o upload estar completo.

PHP e Upload de Arquivos

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Upload de
Arquivos

- Um resumo de como o PHP lida com o upload de arquivos:
 - Formulário HTML
 - Superglobal \$_FILES
 - Script PHP para Upload
 - Segurança

- O primeiro passo é criar um formulário HTML para o upload de arquivos.
- É importante que o formulário tenha o atributo 'enctype' definido como 'multipart/form-data'.
- Isso permite que o formulário envie arquivos binários (como imagens, documentos PDF, etc.) e dados de formulário.

- Quando um arquivo é enviado através de um formulário HTML, as informações sobre esse arquivo são armazenadas na superglobal \$_FILES.
- Isso inclui o nome do arquivo, o tipo MIME, o tamanho do arquivo, e outras informações.

- Você precisa escrever um script PHP para processar o upload de arquivos.
- Este script irá acessar os dados do arquivo através da superglobal `$_FILES`, e então mover o arquivo para o local desejado no servidor.

- É crucial validar e sanitizar os arquivos carregados para evitar problemas de segurança.
- Isso pode incluir a verificação do tipo de arquivo, a verificação do tamanho do arquivo, a renomeação do arquivo, e outras medidas de segurança.

Formulário para envio de arquivos

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Upload de
Arquivos

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <title>formulário</title>
5 </head>
6 <body>
7     <form action="." method="post" enctype="multipart/form-data">
8         Documento: <input type="file" name="file"><br>
9         <input type="submit" value="Enviar">
10    </form>
11 </body>
12 </html>
```

- A superglobal \$_FILES em PHP é uma variável automática que está disponível em todos os escopos de um script. Ela é usada para acessar informações sobre arquivos que foram enviados através de um formulário HTML.

- Aqui está a estrutura da superglobal \$_FILES:

```
1 Array (  
2     [file1] => Array (  
3         [name] => MyFile.txt // 0 nome original do arquivo no computador do usuário  
4         [type] => text/plain // 0 tipo MIME do arquivo  
5         [tmp_name] => /tmp/php/php1h4j1o // 0 nome temporário do arquivo no servidor  
6         [error] => UPLOAD_ERR_OK // 0 código de erro associado a este upload de arquivo  
7         [size] => 123 // 0 tamanho, em bytes, do arquivo enviado  
8     )  
9 )
```

- A chave 'error' pode ter um dos seguintes valores:
 - **UPLOAD_ERR_OK**: Não houve erro, o upload foi bem-sucedido.
 - **UPLOAD_ERR_INI_SIZE**: O arquivo enviado excede a diretiva **upload_max_filesize** em php.ini.
 - **UPLOAD_ERR_FORM_SIZE**: O arquivo enviado excede a diretiva **MAX_FILE_SIZE** especificada no formulário HTML.
 - **UPLOAD_ERR_PARTIAL**: O arquivo foi apenas parcialmente carregado.
 - **UPLOAD_ERR_NO_FILE**: Nenhum arquivo foi enviado.
 - **UPLOAD_ERR_NO_TMP_DIR**: Falta uma pasta temporária.
 - **UPLOAD_ERR_CANT_WRITE**: Falha ao gravar o arquivo em disco.
 - **UPLOAD_ERR_EXTENSION**: Uma extensão do PHP interrompeu o upload do arquivo.


```
1 <?php
2 if (isset($_FILES['file'])) {
3     $uploadDir = 'uploads/';
4     if (!is_dir($uploadDir)) {
5         mkdir($uploadDir, 0777, true);
6     }
7
8     $filePath = $uploadDir . basename($_FILES['file']['name']);
9     move_uploaded_file($_FILES['file']['tmp_name'], $filePath);
10    echo "Upload realizado com sucesso!";
11 }
12 ?>
```

Explicando o documento mostrado anteriormente:

Entre as linhas 3 e 6 verifica se existe a pasta **uploads/**, se não existir cria a pasta e atribui as permissões para acessar a pasta.

Em seguida na linha 8 seta o caminho completo para onde salvar o documento, utilizando o nome da pasta juntamente com o nome do arquivo da variável `$_FILES`.

Depois usa-se a função `move_uploaded_file` para mover o arquivo temporário de `$_FILES` para a localização indicada anteriormente.

O documento mostrado anteriormente não possui nenhuma verificação de segurança, portanto permitindo que qualquer tipo de documento seja enviado e inclusive não faz nenhuma verificação se o documento foi mesmo salvo.

Existem diversas maneiras de aumentar a segurança durante o upload, algumas delas são:

- Limitar tipos de arquivos
- Verificar o Tamanho do Arquivo
- Renomear o Arquivo
- Verificar o Conteúdo do Arquivo
- Armazenar Arquivos Fora do Diretório Público
- Definir Permissões de Arquivo
- Verificar Erros de Upload

- Verifique a extensão do arquivo e o tipo MIME para garantir que apenas tipos de arquivo permitidos sejam aceitos¹.

```
1 <?php
2
3 $allowedTypes = ['image/jpeg', 'image/png', 'application/pdf'];
4 $fileType = mime_content_type($_FILES['file']['tmp_name']);
5 if (!in_array($fileType, $allowedTypes)) {
6     die('Tipo de arquivo não permitido.');
```

¹Lista com os tipos MIME aceitos em https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

- Defina um tamanho máximo para os arquivos que podem ser enviados.

```
1 <?php
2
3 $maxFileSize = 2 * 1024 * 1024; // 2MB
4 if ($_FILES['file']['size'] > $maxFileSize) {
5     die('Arquivo excede o tamanho máximo permitido.');
```

```
6 }
```

- Renomeie o arquivo para evitar a execução de scripts maliciosos com nomes de arquivos especiais.

```
1 <?php
2
3 $uploadDir = 'uploads/';
4 $fileName = uniqid() . '-' . basename($_FILES['file']['name']);
5 $filePath = $uploadDir . $fileName;
```

- Verifique o conteúdo do arquivo para garantir que ele não contenha scripts maliciosos.

```
1 <?php
2
3 $fileContent = file_get_contents($_FILES['file']['tmp_name']);
4 if (strpos($fileContent, '<?php') !== false) {
5     die('Conteúdo de arquivo malicioso detectado.');
```

```
6 }
```


- Armazene os arquivos em um diretório fora do diretório público do servidor web para evitar acesso direto.
- Crie um script separado para servir os arquivos, garantindo que apenas usuários autorizados possam acessá-los.

- Defina permissões de arquivo restritivas para impedir a execução de arquivos carregados.

```
1 <?php
2
3 chmod($filePath, 0644); // Apenas leitura e escrita para o proprietário, leitura para outros
```

- Verifique se houve erros durante o processo de upload.

```
1 <?php
2
3 if ($_FILES['file']['error'] !== UPLOAD_ERR_OK) {
4     die('Erro no upload: ' . $_FILES['file']['error']);
5 }
```

MUITO OBRIGADO!

Cleber Schroeder Fonseca

<http://ifrs.edu.br/riogrande>
profcleberfonseca@gmail.com
cleber.fonseca@riogrande.ifrs.edu.br